Scr
TJ
212
.C58

University of Windsor
APR 19 1966
Library

COMPUTERS AND AUTOMATION

formerly

THE COMPUTING MACHINERY FIELD

Vol. 2, No. 4 May, 1953

Compiling Routines
... G. M. Hopper

Mechanical Translation
... A. D. Booth

Medical Diagnosis
... Marshall Stone

Published monthly except June and August by
Edmund C. Berkeley and Associates, 36 West 11 St., New York 11, N. Y.

(Reprinted 1961, without advertisements, by Berkeley Enterprises, Inc.,
815 Washington St., Newtonville 60, Mass.)

Association for Computing Machinery. The next meeting of the Association for Computing Machinery will be at Mass. Inst. of Technology, Cambridge, Mass., on Wednesday, Thursday, and Friday Sept. 9, 10, and 11, 1953. The Digital Computer Laboratory at MIT is the host computer group.

On April 17 the Council of the Association elected a new secretary: Mr.E. Bromberg, Inst. of Mathematics and Mechanics, New York University, 45 4th Avenue, New York, N.Y. The Council reelected as treasurer of the Association Mr. R. V. D. Campbell, Burroughs Research Division, 511 No. Broad St., Philadelphia 23, Pa.

Society for Industrial and Applied Mathematics. This society, recently formed in Philadelphia, is interested in the "application of mathematics". It now has over 140 "ordinary" members (individuals) and 5 contributing members (corporations). It has begun to publish a monthly newsletter, and plans to publish a quarterly scholarly journal. The editor of the newsletter is Mr. D.B.Houghton, Franklin Institute, 20th St. and Parkway, Philadelphia 3, Pa.

Manuscripts. We desire to publish articles that are factual, useful, understandable, and interesting to many kinds of people engaged in one part or another of the field of computers and automation. In this audience are many people who have expert knowledge of some part of the field, but who are laymen in other parts of it. Consequently, a writer should seek to explain his subject, and show its context and significance. He should define unfamiliar terms, or use them in a way that makes their meaning unmistakable. He should identify unfamiliar persons with a few words. He should use examples, comparisons, analogies, etc., whenever they may help readers to understand a difficult point. He should give data supporting his argument and evidence for his assertions. An article may certainly be controversial if the subject is discussed reasonably.

Ordinarily, the length should be 1000 to 4000 words, and payment will be $10 to $50 on acceptance. A suggestion for an article should be submitted to us before too much work is done. To be considered for any particular issue, the manuscript should be in our hands by the 1st of the preceding month.

Notice. COMPUTERS AND AUTOMATION is published ten times a year, monthly except June and August, by Edmund C. Berkeley and Associates, 36 West 11 St.,New York, 11, N.Y. Copyright, 1953, by Edmund Callis Berkeley. Subscription rates effective April 1, $4.50 for one year, $8.50 for two years, in the United States and Canada; $5.50 for one year, $10.50 for two years elsewhere. Back copies, $1.25 each if available; see more information on page 29. Entered as second class matter at the Post Office, New York, N.Y.

If your address changes, please notify us giving both old and new addresses, and allow three weeks for the change.

COMPUTERS AND AUTOMATION

Vol. 2, No. 4 May, 1953

ARTICLES

	Page
Compiling Routines by G. M. Hopper	1
Mechanical Translation by A. D. Booth	6
Medical Diagnosis by Marshall Stone	23

REFERENCE INFORMATION

Roster of Organizations (supplement)	9
Who's Who: Section 3 — Not Programming, Not Business (cumulative, A to D)	11
Glossary: Section 2 — C, D, E (cumulative)	17
Automatic Computers — List (supplement)	24

DEPARTMENTS

The Editor's Office	i
Forum	21
Advertising	25

Editor: Edmund C. Berkeley

Advisory Committee: Samuel B. Williams, Herbert F. Mitchell, Jr.

Assistant Editors: Eugene F. Murphy, Neil Macdonald

Contributing Editors: George A. W. Boehm, James Gibbons, Alston S. Householder, Henry M. Paynter, Jr., Fletcher Pratt

Publisher: Edmund C. Berkeley and Associates

COMPILING ROUTINES

by Dr. Grace M. Hopper
Vice President, Society for Industrial and Applied Mathematics, Philadelphia

An often-repeated principle in detective work is: "set a thief to catch a thief." This principle has a new parallel: "set a computer to program a computer." Half a dozen years ago, few would have suspected the rich rewards to be gained from programming a computer to put together programs for itself. And yet this was forecast by the symbolic logician, Dr. A.M. Turing, of the University of Manchester, England. In a paper published about 1936, he showed that if a computing machine is able to modify its own instructions, it can solve any problem capable of numerical solution.

In the design of computing machines up to the present the total number of different single instructions or orders that can be given to the computer ranges from about eight to about eighty. The actual number needs to be large enough to be fairly convenient for human beings to use when instructing or programming the machine, yet small enough to place no undue complexity on the hardware of the machine. This compromise at first glance excludes many special orders which in a few problems could be very useful, such as an order for finding the cube root of a number.

The absence of special orders of this kind is solved by the technique of programming their equivalent and recording them in the memory of the computer for reference, rather than by installing additional hardware. This is possible as soon as the computer is supplied both with an adequate internal primary memory and a rapid and flexible input-output equipment providing a very large secondary memory. This secondary memory must be at the beck and call of the computer itself, completely and at all times, if the computer is to be capable of modifying its own instructions with marked efficiency.

The concept of a computer constructed with a minimum instruction code in hardware, and yet delivered to the customer with a large and flexible set of orders stored in the secondary memory is not new. The computer known as the Harvard Mark III had a "coding machine" and was one of the first to be so constructed. The Remington Rand machine Univac has been supplied with a general code stored on tape. The trend is evident. In the future, customers will express less concern as to whether a computer is "mathematical", "commercial", or "logistic". For the computer will be truly general-purpose, while its library of programs will particularize it as may be convenient for mathematical or commercial or logistic applications. The identical computer may be an engineering instrument at one moment with one program and a pay-roll clerk at the next moment with another program.

The development of a library of subroutines for the machine Edsac has been described by Dr. M.V. Wilkes, of Cambridge University, England. Following the knowledge of his work, an extensive family of program-making programs known as "interpreters, compilers, generators, operators, editors, executives" have been developed for major operating computers. Since there is a current shortage of programmers, and considerable pressure on computer time these developments have been so far somewhat of a hodge-podge with overlaps and gaps.

Consequently, more lines of communication among programmers need to be opened and a comprehensive plan for expansion of program-making programs needs to be laid down. Also, it is desirable that programmers work side by side with logical de-

signers and engineers at the time that the design of a computer, large or small, is begun. Thus, a computer will be delivered with its basic programs tested and proven, ready to be used flexibly and conveniently.

First attempts to reduce programming time were made in the direction of mathematical programs. Here the concept of subroutines is most evident. The elementary algebraic and transcendental functions are clearly defined, easily symbolized, and convenient. In the Harvard IBM Mark I computer, which in 1944 could not compute its own instructions to any great extent, these functions were wired in as relay circuits automatically sequenced by the step-counters which controlled the "multiply-divide" unit of the machine. Faults of such wired-in circuits soon made themselves evident. They always computed to full accuracy. They tied up the transfer buss and hence the entire computer. They took no cognizance of approximations that might be available in the current computation but depended on stored function tables. They evaluated series or iterated to the limits of computer accuracy always for the worst possible case. They were inflexible.

By contrast, the computer able to modify its own program can be directed to evaluate the terms of a series only until these terms became less than an assigned tolerance. Iterative routines can be repeated only until the difference between two successive approximations becomes sufficiently small. No longer is it necessary to write an all-inclusive "straight-line" program; instead an iterative "loop" program with a test for repetition of the loop is used. Subroutines are carefully planned, both to conserve memory space and to operate with as much speed as possible. Naturally, when effort and ingenuity have been expended upon such subroutines, the desire to use them again in another program is strong.

This desire to use again a good subroutine however, immediately poses the problem of locating it in the computer's memory. For an "operative subroutine", i.e., a subroutine which carries out a complete mathematical operation, two courses are available. These lead to the interpretive method and the compiling method.

The interpretive method of using good subroutines consists of fixing the location of the subroutine in the computer memory, and causing the main program to interpret what may be called a "pseudo-code", and thus refer to the subroutine and perform it. Evident or concealed, this procedure, beyond the decoding, involves the following steps: 1 — transfer arguments to the subroutine or to some standard location; 2 — mark position in the main routine; 3 — transfer control to the subroutine; 4 — carry out the subroutine; 5 — transfer control back to the main routine; 6 — retrieve results from the subroutine or standard location. All but step 4 are waste motions if the program is executed more than once.

The compiling method of using subroutines consists of copying the subroutine into the main routine, adjusting memory locations as necessary to position the subroutine properly in the program and to supply arguments and results. This method is subject to all of the errors of human copying until the computer itself is instructed to carry out the copying.

With computer copying and either interpretive or compiling routines, it becomes possible to construct pseudo-codes, in which mathematical functions are represented by single computer words. For example, in Univac the computer word C00 00X 000 00Y means $y = \cos x$, and the computer word E00 00U 00N 00V means $v = u^n$.

Furthermore, no memory locations are now assigned by the programmer. Instead, they are systematically assigned by the interpretive or compiling routine. If an

interpretive routine or interpreter is used, a word is examined, the computer carries out the required operation, and then returns to examine the next word. Since the interpretive routine programs and computes simultaneously, it is well adapted for problems which need to be repeated. But if a compiling routine or compiler is used, when a word is examined, the required subroutine is transcribed, suitably modified, into a running program. Thus, after compilation, all the reference steps are eliminated and only step 4 may need to be repeated — the desired operation.

Hence, for a program to be repeated, the compiling method offers a saving of time. The decision to use an interpreter or a compiler however depends upon the particular computer and the number of times the program is to be repeated; and the choice to be preferred can be evaluated mathematically.

In both cases, the advantage over manual programming is very great, once the basic subroutines have been tested and proved. The saving of time for a compiler is usually greater. In fact, the reduction in programming time for those classes of problems upon which compilers have been tested is incredible, actually little short of fantastic. Programming can be reduced from weeks to a matter of minutes.

This enormous reduction depends on developing compilers which are general and intelligent, and which may be called executive routines. They need to be devised, tested and supplied with numerous volumes of good subroutines in their libraries. The use of interpreters and compilers also eliminates the time spent in de-bugging programs, provided of course that the original statement of the problem is correct, and the computer and its input-output equipment are suitably checked. For instance, it is possible to direct the compiler to refer to a volume of floating-decimal, double-precision, or complex number subroutines. Not so obvious but essentially simple extensions are to direct the compiler to program for a certain accuracy, to scale quantities as necessary, to supply information on round-off errors. The libraries can be made interchangeable, and compilers can be designed to program for any computer. A problem stated in a basic pseudo-code can thus be prepared for running on one or more computers if the corresponding compiler and subroutine library are available. But so far this has actually been carried out only for one computer and only in prototype form; this step however has proved the feasibility.

It has been found that the use of a pseudo-code imposes certain valuable standards on the statement of the requirements of a problem and serves as a check on the mathematician. In general it requires the programmer to list: operation required; arguments; results; next operations. This forms a sort of four-multi-address code. For example, if it is desired to integrate $I = \int_a^b y\,dx$ by Simpson's Rule, it is necessary to state: — operation: integrate, Simpson's Rule; — arguments: y, a, b, Δx; — result: I; — next operation: ..... The quantity Δx may be specified by the programmer, or the compiler can be constructed to select Δx on the basis of certain information. It is best to instruct the compiler to use Δx as supplied or if not supplied to determine an appropriate value. It has been found advantageous to supply the compiler with a reasonable answer, and a method of indicating its decision, when information is missing. This implies long and careful work on the part of the programmer preparing the executive routines and subroutines. The examples cited may seem trivial but it has been found necessary to proceed slowly and carefully from the trivial to the complex in producing the executive routines.

The mathematical problem has been comparatively easy to state. With its vast symbolism, commonly understood through long usage, mathematics possesses an exact language readily embedded in compilers and subroutines. This is not true of commercial routines, nor even of business arithmetic. Even here however, some subroutines are immediately evident, for such procedures as income tax deductions,

city wage taxes, and social security deductions. These lead to an attempt to devise a symbolism. If "cos" indicates cosine, why not use "foa" for "federal old age" or social security deduction? The prototype pseudo-code so far developed is redundant, and depends heavily on plain English. However, it is proving to be a useful vehicle for exploration.

In the commercial routines moreover, the number of arguments and results multiplies rapidly. Consider a skeleton pay roll routine. For federal income tax, it is necessary to know for each individual, his gross pay, and the number of his dependents. For city wage tax for each individual, it is necessary to know his gross pay, whether the individual is subject to tax (i.e., resident or non-resident), and his job, and for each job, whether the job is subject to tax (i.e., within or without the city boundaries). For social security deduction for each individual, it is necessary to know gross pay and accumulated pay. However, the task is by no means impossible. Perhaps sufficient attention has not been focused on the fact that in defining a mathematical problem, it is simple to assume that double-precision subroutines or floating-decimal complex number subroutines are available. It seems that it will be necessary to standardize a pay-roll notation in order to use a pay-roll compiler and a subroutine library.

The objection that most payrolls will not be presented in the standard form can be met by supplying the computer with what may be called editing routines. Some editing routines will simply transcribe raw data into a standard form suitable for computer processing. Other editing routines transform computer data, supplying typewriter characters, into a form suitable for printing on reports, bills or checks. To construct editing routines, what may be called generating routines have been devised. The generating routine is supplied with specifications listing the position of information in the raw data and its desired position in the smooth data such as headings and columnar positions. The generating routine, given these specifications, computes and records an editing routine. Moreover, the resulting program is extremely fast, since the computer can be instructed to select the most advantageous allocation of the memory, input, and output, depending on size of item and number of operations to be performed. Editing is but one phase of the commercial and logistic problems which lend themselves to generator techniques.

Turning once more to the mathematical field, some progress is being made in devising routines corresponding to mathematical operators. The first compiling routine to be tested and proved is the differentiator, and corresponds with the mathematical operator D (for derivative). If a function is presented, implicitly or explicitly in the pseudo-code (referring to the basic subroutine libraries), the operator routine for D will guide the computer to produce the analytical formulas for the nth derivative, or the first n derivatives, or the nth through mth derivatives, as may be specified. Such derivatives may be complete or partial, of any order up to and including the ninety-ninth (at present writing). The formulas derived by an operator routine can be immediately submitted to an interpreter or compiler to produce an error-free program for numerical evaluation. In this way, much has been learned about computer manipulation of algebraic rather than numerical data. It would be foolhardy to attempt to predict the limits of possible further development in this field.

Throughout this discussion the statement of exact timing has been avoided. Some of the executive routines proved and tested have been prototypes in which speed has not been the primary consideration; yet in spite of this, the ratio of total programming and computer time using executive routines to total programming and computer time using man-made programs is almost incredible. Finally, the library of executive routines and subroutines is as yet far from complete; and only certain classes of problems can be readily treated. It is worth saying again that

compiling routines are applicable to any computer with adequate checking facilities and a sufficiently large secondary memory, ranging from 2400 to 50000 single address orders.

The programs proved and computer-tested to date, consist of: two forms of compilers; floating-decimal library; double-precision library; part of a double-precision floating-decimal library; special and general editing generators; the differentiator. Much has been accomplished on a pay-roll compiler, but it is still several months from completion.

This work is necessarily group research, and this account cannot be published without citing those members of the programming development group primarily responsible for the achievement of these results: Richard K. Ridgway, John Waite, Adele Mildred Koss, Harold E. Sweeney, Harry Kharimanian, Frank Delaney, James McGarvey, and Margaret Harper. All of the work was carried on under the supervision of Dr. Herbert F. Mitchell, Jr., and Richard D. Woltman, whose criticisms and encouragement have proved very valuable.

MECHANICAL TRANSLATION

by Andrew D. Booth
Birkbeck College, London, England

During the summer of 1947 I first suggested that a digital computer having adequate memory facilities could perform the operations necessary to translate a text written in a foreign language (F L) into the desired language or target language (T L). There was, and is, no particular difficulty in doing this, as I hope to show in the present article; but I make no claim that a literary quality in the result of the translation is to be hoped for.

The original proposals covered only the making of a straightforward dictionary translation from the foreign language to the target language. It is convenient to start by seeing how this simple objective may be achieved on a machine whose primary purpose is the manipulation of numbers. It is necessary to assume only the most rudimentary machine functions in order to perform mechanical translation (M T):

a) The machine has a large memory.

b) The input typewriter sends data, either direct to the memory, or to a register provided with subtraction facilities, the accumulator register.

c) The machine contains a conditional transfer order which enables the machine to select between alternative courses of action according to the sign of the number held in the accumulator register.

d) The contents of the accumulator can be typed at the output.

The reader familiar with modern automatic digital computers will see that all of the above functions are present in all such computers existing, with the exception in many cases of the large memory.

How shall we represent the foreign language text in digital form? A normal teletype machine is so constructed that the depression of any key, for example that corresponding to letter A, causes the emission of a binary coded digit pattern which has a one-one correspondence to the desired character. Thus:
A becomes 00011; B becomes 11001; C becomes 01110; ....; and Z becomes 10001,
It follows that, if the keys corresponding to the letters of the foreign word are depressed, in sequence, a digital pattern will be generated which uniquely represents that word. If this pattern is regarded as a number, a dictionary translation of the foreign word can be obtained by storing the translation in that memory location which has the same number as the code of the foreign word. As an example, the Latin word et is coded 10000, 00001, which as a binary number is equal to 512 plus 1 or 513 and would identify memory location 513. Then in that memory location 513 we would store the translation: 00100(d), 01100(n), 00011(a) corresponding to "and". The reader interested in details will notice that it is assumed that digits are shifted into the machine register, starting from the least significant (right shift), and that the inversion of order (d, n, a) is necessary for the output type to appear in the normal sequence.

It is at once obvious that this simple scheme is quite impracticable, since even in the example given, it will be seen that 1024 locations are required to deal even with the two letter words of the foreign language. For words of maximum length say 10 letters, 2^{50} locations would be needed. This would exceed even the most sanguine hopes of modern machine designers. In any case no known foreign language has anything approaching 10^{15} (2^{50}) words, so that almost all of the memory would be empty.

The difficulty is easily overcome, however. Suppose that each location (in sequence) in the memory contains a "dictionary" word (D W) having the following composition: the F L word (10 letters say) and the T L translation (40 letters say). Assume that the D W's are stored in ascending order of magnitude. Then if the F L word is subtracted from each of the D W's in turn, the result will be negative until the required entry is reached and positive thereafter. It follows that, if the conditional transfer is used to break off the sequence of subtractions at the first positive result, the remainder in the accumulator at this point will represent the target language translation. The latter may now be printed at the output.

A second obvious point is that the length of the required words (250 binary digits or bits in the above example) is considerable. Existing computing machines fall short of dealing with this by a factor of five or greater. They may, however, easily be programmed to use multiple length words so that this is not an essential difficulty.

If the actual F L word is not contained in the memory, the nearest equivalent will be generated by the above process. Furthermore, since the D W, F L entry will be numerically somewhat larger than the text F L word, the output operation will generate certain nonsensical characters before the T L translation. This will indicate to the reader that an untranslatable word is present.

The preceding simple scheme is much limited by the available memory in existing (and near future) machines. But in 1948 R. H. Richens suggested to me a modification which makes mechanical translation a really practicable operation. Richens pointed out that, with certain limitations, an adequate or passable translation of a foreign language text would result from the following operation:

a) The memory contains a stem (or root) dictionary and an ending dictionary.

b) The stem dictionary consists of a relatively few entries of general semantic utility plus a vocabulary specific to the subject of the translation.

(The latter has since been called, by V. Oswald, Micro-semantics).

The method of operation is simple. First the F L word is subtracted in turn from the entries in the stem dictionary. In this way, the longest possible stem entry is found. At this point the stem translation and suitable grammatical notes are typed out. The stem is now removed from the F L word, and the remainder is compared with the entries of the ending dictionary. When coincidence is attained again, the relevant syntactic information, contained with the ending entry, is typed out.

Richens has shown that the same method can be applied to mutliple words of the type encountered in, for instance, German.

As an example of this procedure consider the translation of the Latin word amo. This would proceed as follows:

Stem: Trial 1: a, alas
Trial 2: am, love (v) (v for "verb")

Ending: Trial 1: o, (1.s.p.) (for "1st person singular, present tense")

The total output would be: love (v) (1.s.p.)

Certain difficulties arise, as in the example desideremus given by Richens. Here two possible translations exist: (1) desider, desire; emus (1.p.s.a.); or (2) desid, be idle; eremus (1.p.i.s.a.). Resolution could be attained by storing the word itself, together with both translations.

Again, certain words, or parts of words, are sometimes without significance, for example the t in the French a-t-il. In this case, to avoid confusing the operator, the machine probably would have to put out some encouraging symbol, such as "N" for no significance.

It has been suggested, by Prof Erwin Reifler of the Univ. of Seattle, Washington, that semantic ambiguities could be considerably eliminated by the use of a person called a "pre-editor" who could be a native in the F L but would not necessarily know the T L at all. The duty of the pre-editor would be to replace all ambiguous words by non-ambiguous equivalents.

The foregoing brief account of mechanical translation is naturally incomplete in many respects. The act of coding a given example for a particular computer involves many points which it has been impossible to cover in a short article. This is particularly true of the stem-ending dictionaries, whose use requires a high degree of sophistication in the program if a good working speed is to be attained.

Some of these problems however have been actually examined on our computer APEXC at Birkbeck College, London, and the reader may be interested in the following statistics:

Time taken to translate a 1000 word message by a skilled bilingual human being	1 hr.
Time of mechanical translation using the above technique, on standard punched card equipment	1 hr. 54 mins.
Time of mechanical translation on APEXC using teletype output	2 hrs. 15 mins.
Time of mechanical translation on APEXC with tabulator output	30 mins.

It does not appear likely that with existing input-output equipment any much greater speed is possible. The translations, produced by the above methods are of course inelegant, but are easily understood by a person expert in the subject of the paper. Neither the present author nor Richens envisage the literary use of mechanical translation in the near future or even foreseeable future; but within its limitations, the method should be of great use to students and institutions confronted with the mass of published material in foreign languages which is currently appearing.

ROSTER OF ORGANIZATIONS IN THE FIELD OF COMPUTERS AND AUTOMATION

(Edition 8, supplement, information as of May 1, 1953)

The purpose of this Roster is to report organizations (all that are known to us) making or developing computing machinery, or components, or data-handling equipment for automatic control and materials handling. Each Roster entry when it becomes complete contains: name of the organization, its address, nature of its interest in the field, kinds of activity it engages in, main products in the field, approximate number of employees, year established, and a few comments and current news items. When we do not have complete information, we put down what we have. The term "components" as used here does not include, nuts, bolts, resistors, condensers, motors, tubes, mercury, etc., but does include magnetic drums, cores, tapes, and certain other components that have an intimate and significant connection with machinery covered in the Roster.

We seek to make this Roster as useful and informative as possible, and plan to keep it up to date in each issue. We shall be most grateful for any more information, or additions or corrections that any reader is able to send us.

Although we have tried to make the Roster complete and accurate, we assume no liability for any statements expressed or implied.

This edition contains only revisions or additions as compared with Edition 7, cumulative, published in the April, 1953, issue of COMPUTERS AND AUTOMATION, vol. 2, no,2.

Abbreviations

The key to the abbreviations follows:

Size

- Ls Large size, over 500 employees
- Ms Medium size, 50 to 500 employees
- Ss Small size, under 50 employees
  (No. in parentheses is approx. no. of employees)

When Established

- Se Organization established a short time ago (1942 or later)
- Me Organization established a "medium" time ago (1923 to 1941)
- Le Long established organization (1922 or earlier)
  (No. in parentheses is year of establishment)

Interests in Computers and Automation

- Dc Digital computing machinery
- Ac Analog computing machinery
- Ic Incidental interests in computing machinery
- Sc Servomechanisms
- Cc Automatic control machinery
- Mc Automatic materials handling machinery

Activities

- Ma Manufacturing activity
- Sa Selling activity
- Ra Research and development
- Ca Consulting
- Ga Government activity
- Pa Problem-solving activity
- Ba Buying activity
  (used also in combinations, as in RMSa, "research, manufacturing and selling activity")

*C This organization has very kindly furnished us with information expressly for the purposes of the Roster, and therefore our report is likely to be more complete and accurate than otherwise might be the case. (C for Checking)

*A This organization has placed an advertisement in this issue of COMPUTERS AND AUTOMATION. For more information, see their advertisement. (A for Advertisement)

Alden Electronic and Impulse Recording Equipment Co., Alden Research Center, Westboro, Mass. *C, *A
Facsimile recording equipment and facsimile components. Ma SEE Alden Products Co.

Alden Products Co., 117 No. Main St., Brockton, Mass. *C, *A
General and specific components for digital and analog computing machinery; plug-in components, sensing and indicating components, magnetic delay line units, magnetic storage cores, etc. Ms(300) Me(1930) Ic RMSa

Alfax Paper and Engineering Co., Alden Research Center, Westboro, Mass. *C, *A
Electrosensitive recording papers. Ma SEE Alden Products Co.

Curta Calculator Co., 3851 West Madison St., Chicago 24, Ill.
Eight-ounce, hand-powered, rotary "brief-case" calculator; adds, subtracts, multiplies, divides; totals to 11 decimal places; made in Lichtenstein. Dc Sa

Davies Laboratories, Inc., 4705 Queensbury Road, Riverdale, Md.
ANSER, analog simulator and computer. Ac Ma

Elliott Addressing Machine Co., 199 Albany St., Cambridge 39, Mass.
Addressing stencils, with selection controlled by punched holes in cardboard margin. Ls Le Ic RMSa

E-Z Sort Systems, Ltd., 45 Second St., San Francisco 5, Cal.
Edge-punched cards for filing and sorting data. Ic

Fabrica Addizionatrice Italiana S.A., Viale Umbria 36, Milan, Italy
Desk calculators; add, subtract, multiply, divide, print. Dc RMSa

Hamann Calculating Machine Co., 2118 Land Title Building, Phila. 10, Pa.
Adding, subtracting, multiplying desk calculators. Dc MSa

Hughes Research and Development Laboratories, Hughes Aircraft Co., Culver City, Calif. *A
Automatic data handling systems. Industrial process control systems. Small powerful electronic digital computer. Fire-control equipment. Aircraft control. Navigation systems. Ls Me DAc RMSa

Intelligent Machines Research Corp., 134 So. Wayne St., Arlington, Va. *C, *A
Devices for reading characters on paper, etc. Pattern interpretation equipment. Sensing mechanisms. Digital computer elements. Ss(6) Se(1951) Dc RCMSa

International Business Machines Corp., 590 Madison Ave., New York 22, and elsewhere. *C, *A
Punch card machines. IBM Electronic Data Processing Machines, Type 701 (magnetic tape, magnetic drum, electrostatic storage). Card Programmed Calculator. Electronic calculating punch Type 604. Data processing equipment. Process control equipment. Automatic Source Recording Equipment. Ls(42,000) Le(1911) Dc RMSa

Lanston Monotype Machine Co., Barrett Adding Machine Div., 24th & Locust Sts., Phila. 3, Pa.
Adding, subtracting, and printing, desk calculators. Dc RMSa

Monrobot Corp., Morris Plains, N.J. *C, *A
Electronic digital computers. Adding, calculating, bookkeeping and electronic digital computers; Monrobots. Subsidiary of Monroe Calculating Machine Co. Ss(32) Se(1953) Dc RMSa

Monroe Calculating Machine Co., Orange, N.J. *C
Desk calculating machinery. SEE Monrobot Corp. Ls(4000) Me(1925) Dc RMSa

George A. Philbrick Researches, Inc., 230 Congress St., Boston 10, Mass. *C, *A
Philbrick electronic analog computing equipment and components. Ss(5) Se(1946) Ac RCMSa

WHO'S WHO IN COMPUTERS AND AUTOMATION: SECTION 3

— NOT BUSINESS, NOT PROGRAMMING — A TO D

(First edition, cumulative, information as of April 25, 1953)

This is a fourth installment of a Who's Who of individuals in the field of computers and automation. The purpose of this Who's Who is to make it easier for all persons interested in this field to get in touch with each other in appropriate ways.

Contents. The following list consists of persons interested in computing machinery who have not reported as a main interest either "programming" or "business" for whom information has been received up to April 25, 1953, and whose last name begins A to D.

Reporting. If you are interested in any phase of computing machinery, robots, cybernetics, or automation, and if you would like to be included in the Who's Who, please send us: your name (please print), address, organization (and its address), your title, main interests (note list appearing under "Entry" below, and specify any other interests), year of birth, your college or last school, years of experience in the field, your occupation, and any more information about yourself that you may care to furnish. (A card for your Who's Who entry is bound in with this issue.) Your listing in the Who's Who does not depend in any way on your subscribing to COMPUTERS AND AUTOMATION although of course your subscription will be welcome.

Entry. Each entry in the Who's Who when it becomes complete contains: name / title, organization, address / interests / year of birth, college or last school (background), years in field, occupation. The address has been substantially contracted to avoid the nuisance of unwanted mail. In cases where no information has been given (for example, about occupation) a "-" denotes omission.

Abbreviations. Since a great deal of information is to be presented, abbreviations have been extensively used. Nearly all these abbreviations can be easily guessed, like those in a telephone book. The letters A,B,C,D,E,M,P,S stand for main interests "Applications, Business, Construction, Design, Electronics, Mathematics, Programming, Sales" respectively.

Liability. Although we have tried to make each entry complete and accurate, we assume no liability for any statements expressed or implied.

Corrections. We shall be very grateful for any information, additions, or corrections that any reader is able to send us.

Roster

A: Abramson, Norman / stud, Harvard, Camb, Mass / ADEM / '32, Harv, 0, stud
Ackerlind, Erik / pres & tech dir, The Wallind-Pierce Corp, Lomita, Cal / ACDEM / '10, Polytech Inst Bklyn, 6, elecncs engr
Adler, Cyrus / engr, Western Elec, N Y / ADEM / '27, Inst for Math NYU, 2, mathm
Aghib, Edward G / meth engr, Rem Rand, N Y / A, sys plng / '22, IEPolytech Inst Univ Lausanne, 10, meth engr

Ahbol, Kenneth W / dir, Engrg & Des Div, US Naval Stat, Puerto Rico / ACDEM/'13, CCNY, 16, civ & mech engr

Allen, Bonnell H / mgr, Engrg Soc Personnel Serv, Chic / personnel / '04, -,-,-

Allen, Jack K / res asst, WRRC, Ypsilanti, Mich / ACDEM / '24, Wayne Univ, 1, elec engr

Allen, William R / grp leader (mathn), Advisory Bd on Sim, Univ Chic, Chic / AM, statistics / '20, Nwn U, 1, mathn & statn

Ambrosio, B F / asst chf, Mach Dev Unit, Natl Bur Std, Inst Num Anal, UCLA, Los Angeles / ACDE / '13, U Mich, 15, elecnc engr

Ammerman, Donald W / dev engr, Rem Rand, Conn / CDE / '20, Wash State, 6, engr

Anderson, Frederick J / hd, Comp Grp, Sylvania Elec, Boston / AE / '23, Stanford U, 5, elec engr

Anderson, L L / mgr, Magnetic Eqpt Dept, Brush Devt Co, Cleveland / S / '18, U Louisville, 3, engrg sales

Anderson, Walter L/staff elecl engr, Engrg Res Assoc, Arlington, Va / AE/ '22, U Minn, 6, elec engr

Andrews, D R / dev engr, RCA, Camden, N J / dev engrg on input-output sys / '07, Ball State Teachers Coll, 3, engr magn recording

Angell, Worcester Randolph, Jr / mathn, Perkin-Elmer Corp, Camb, Mass / M, optics / '27, Tufts Coll, 1, mathn

Arenberg, D L / owner, Arenberg Ultrasonic Lab, Jamaica Plain, Mass / CD / -, MIT, -, physicist

Arnold, Kenneth J / assoc prof, Mich State Coll, East Lansing, Mich / A, statistics / '14, MIT, 7, teacher

Arthur, Paul, Jr / member, Amer Chem Soc Ctee, Wilmington, Del / A, literature & infm handling / '15, MIT (PhD'38), 0, res chem

Asmuth, John L / instructor, Dept Elecl Engrg, U Wisc, Madison / CDE / '22, U Wisc, 3, teaching elecl engrg

Avakian, Emik A / consltnt, IBM, N Y / A, infmn processes / -, -, -, consltnt

Avery, Harold T / chf engr, Marchant Calculators, Inc, Oakland, Cal / AD, mech. calculators / '92, Stanford U, 24, engr

Ayres, William R / engr, RCA Victor, Camden, N J / E, sys & circuit dev / '18, Kans (BSEE'39), 2, engr

B: Balakian, George / dev engr, Eckert-Mauchly Comp Corp, Phila /E / '22, Rutgers U, 1, elec engr

Balleisen, Charles E / supvr, Southwest Res Inst, San Antonio, Tex / CD, comptg linkages / '11, MIT, 5, res admnr

Baldwin, Woodson W / engr, Rand Corp, Los Angeles, Cal / A / '16, MIT, -, engr

Bangert, John T / tech staff, Bell Tel Labs, Murray Hill, N J / AM / '19, U Mich, Stevens, Columbia, 3, elecl engr

Baran, Monroe / elecnc engr, Western Elec Co, Dayton, O / DE / '21, Cooper Union Sch Engrg, 6, field engr navigation sys

Barrows, John C / comptroller, Amer Surety Co, N Y / A / '09, Harv Sch Bus Admn, 20, comptroller

Barry, James E / res asst, WRRC, Ypsilanti, Mich / ADM, log des / '29, Wayne U, 3, -

Bastuscheck, Clifford P / physicist, Haller Raymond & Brown, State College, Pa / CD / '23, Pa State Col, 2, physicist

Baugh, Harold W / elecnc sci, NAMTC, Pt Mugu, Cal / ACDE, Raydac / '24, Cal Tech, 4, elecnc engr

Beagan, Walter J / fld engr, Consol Engrg Corp, N Y / AS / '19, Pratt Inst, 2, engr

Beer, Albert C / princ physicist & asst suprvr, Battelle Memorial Inst, Columbus, O / AM / '20, Cornell U, 4, physicist

Begun, S J / -, Brush Dev Co, Cleveland, O / memory dev / '05, -, 5, exec engr
Belcher, Harold P / elecnc engr, Natl Bur Std, Wash, D C / ACDE / '26, Howard U, 2, elecnc engr
Bell, J C / asst suprvr, Battelle Memorial Inst, Columbus, O / M, appld math / '15, U Ill, 3, mathn
Bell, William D / vp, Telecomputing Corp, Burbank, Cal / M / '20, UCLA, 12, -
Bellinger, James E / -, Patrick Air Force Base, Cocoa, Fla / D / '25, Rensslr Poly Inst, 1, mechl engr
Belsterling, Charles A / res asst, Franklin Inst Labs, Phila / ACDEM / '25, Drexel Inst, U Pa, 1, elecl engr
Bemis, Philo S / sys engr, Amer Dist Telegraph Co, N Y / AM, log des, mathl stat, switching circ / '15, U Portland (BS), Stevens, 3, engr
Benton, Robert F / sys analyst, NAMTC, Pt Mugu, Cal / A, data reduction sys/'22, Cal Tech, 3, elecnc sci
Berlinsky, Anthony A / des engr, Bur Census, Wash, D C / ACD / -, Geo Wash U, 12, des engr
Berry, H W / res engr, Aeroncs Div, Mnpls Honeywell Regulator Co, Mnpls, Minn / ADE / '12, U Pa (MS), 3, res engr
Bianco, Richard J / elecnc sci, Comp Lab, BRL, Aberdeen Prvg Gr, Md / ADE, log des / '22, Iowa State Coll (BSEE), 3, des engr
Bialo, John M / elecncs analyst, Freed Radio Corp, N Y / EM / '22, Stevens Inst Tech, NYU, 4, engr
Bishop, John F / asst gen mgr, Beckman Instr, So Pasadena, Cal / - / '24, Harv Bus Sch, -, -
Blachman, Nelson M / physicist, Comp Br, Ofc Nav Res, Navy Dept, Wash, D C/ ADM, comp, comp compnts / '23, Harv U (PhD), 2, physicist
Bloch, Alan / vp & chf engr, Audio Instr Co, N Y / DE, sys stud / '15, Oberlin, Iowa State, 7, physicist
Blythe, John H / navign sci, USN Hydrographic Ofc, Chevy Chase, Md / M / '99 , Geo Wash U (MA), -, govt sci
Blythe, Richard H, Jr / operns analyst, Air Defense Comm, Colo Springs / A / -, Yale, -, statn
Boehm, George A W / sci ed, Newsweek, N Y / M / '22, Columbia U, 8, editor
Boelter, L M K / dean, Coll Engrg, Univ Calif, Los Angeles / A / '98, -, -, instructor
Bonn, Theodore H / proj engr, Rem Rand, Eckert-Mauchly Div, Phila / D, solid state phys, new storage dev / '23, U Pa, 5, elecnc engr
Borders, C R / techl engr, IBM, N Y / CD / '22, U Cincinnati, 4, dev & des engr
Boulanger, Eugene / asst suprvr, Planning, Mutual Life Ins Co, N Y / A / '04, -, -, -
Boyer, Ray W / proj engr, Royal Elec, Dayton, O / ACD, statcl accelerometer/'11, Miamisburg, 20, engr
Boynton, Ralph E / asst dir staff trng, Bank of Amer, Los Angeles, Cal / AE, banking / '16, Occidental Coll, -, -
Brainerd, J G / prof, Moore Sch Elec Engrg, U Pa, Phila / general / '04, U Pa , 15, prof
Bramble, Charles Clinton / dir res, USN Prvg Grnd, Dahlgren, Va / AM / '90, John Hopkins U, 10, -
Brandt, A E / biometrician, Health & Safety Div, AEC, N Y / AD / -, Iowa State Coll, 25, -
Braun, Edward Louis / dev engr, Librascope, Glendale, Cal / ADEM / '25, U Cal, 4, dig comp desgr
Bridge, William H / elecnc sci, Elecnc Comp Lab, Natl Bur Std, Wash, DC / ADE / '27, Cath U of Amer, 2, desnr

Briggs, Thomas H / dept mgr, Burroughs Adding Mach Co, Phila / ACDS / '06, Cal Inst Tech, Wesleyan U, 3, engr

Broadwell, Herbert S / dev engr, Adv Dev Sec 1018, RCA, Camden, N J / ADE / '21, CCNY, 2, engr

Brooke, A Wayne / techl engr, IBM, Endicott, N Y / CDE / '13, Case Inst Tech, 15, engr

Broomall, John M / elec engr, Mnpls Honeywell Brown Instr Div, Phila / E, contr sys / '16, Moore Sch U Pa, 10, engr

Brown, David R / engr, Dig Comp Lab, MIT, Camb, Mass / DE, compnts / '23, U Cal, 6, elec engr

Brown, Donald Meeker / suprvr, Math Serv Grp, WRRC, Ypsilanti, Mich / M / '07, U Ill, 2, engr

Brown, R G / hd, Operns Res Dept, WRRC, Ypsilanti, Mich / ADM / '23,-(MA), 5, operns analyst

Brown, William G / res assoc, WRRC, Ypsilanti, Mich / DM, log des / '29, U Mich, 2, -

Brownell, H Russell / chf elec engr, Magnetic Metals Co, Camden, N J / D, res on static mag mem / '09, Penn State, 2, res & dev engr

Buck, Dudley A / res asst, Dig Comp Lab, MIT, Camb / CD, solid state dev for mem & switching / '27, MIT, 3, -

Buell, Elliott L / mathn, Aerial Measurements Lab, Nwn U, Evanston, Ill / AM / '16, MIT, 3, mathn

Bunzel, David G / TV engr, Standard Coil Prod, Los Angeles, Cal / DEM / '24, Amer Tel Inst Tech, Chic, 0, elecncs engr

Burington, Richard S / -, Bur Ord, Navy Dept, Wash, D C / AM, sys evaln, physics/ '01, Ohio State, 9, mathn

Burns, Loren V / owner, Burns Assoc, Manhattan, Kansas / A / '12, Kans U, 6, consltnt in feed technology

Burrell, Warren P / sr elec engr, Engrg Res Assoc, St Paul, Minn / DE, dig comp sys, infmn proc sys / '23, U Minn, 5, engr

Burstein, Albert / asst res engr, Burroughs Adding Mach Co, Phila / DE / '21, U Pa, 3, engr

Butler, J W / -, Argonne Natl Lab, Chicago / AM / -, -, -, mathn

Butler, William W / proj suprvr, Engrg Res Assoc, St.Paul / AD / '19, Iowa State Coll, Cal Inst Tech, 5, elec engr

Byrne, Robert M / dev engr, Goodyear Aircraft Corp, Akron, O / ACD, automatic contr / '21, MIT, 5, engr

C: Cabaniss, Edward H / engr, General Elec Co, Schenectady, N Y / DM, dev analog comp / '16, MIT, 7, engr

Cahn, Albert S, Jr / asst to dir of res, Natl Bur Std, Inst for Num Anal, Los Angeles, Cal / M / '10, U Chic, 5, -

Caldwell, S H / prof elec engrg, MIT, Camb / DE, tchg / -, MIT, 23, tchr & elec engr

Callan, Margaret M / asst res offcr, Mech Engrg Div, Natl Res Council of Canada, Ottawa / AM, aerodynamics / '21, U Tor, 10, apld mathn

Carroll, Joel / geodetic mathn, US Geol Survey, Topographic Div, Wash, D C / M / '18, Geo Wash U, -, mathn

Carter, William M / servomech engr, Bell Aircraft Corp, Buffalo, N Y / AM, anal comp / '27, W Va U, 1, engr

Cary, A F / mgr, L & O Res & Dev Corp, Phila / CDE, anal comp dev / '18, Temple U, 8, physicist

Casciato, Leonard / res engr, Compn Centre, U Toronto, Canada / ACDEM / '27, U Tor (MA), 3, res engr

Chalmers, R G / staff, Elecncs Sec, Natl Physical Lab, Middlesex, England / AEM / '09, London U, 5, -

Chambers, F T / proj engr, Applied Sci Corp, Princeton, N J / ACD / '24, Princeton U, 3, elecnc engr

Chapline, Joseph D, Jr / chf, Edit Dept, Eckert-Mauchly Div, Rem Rand, Phila / AC, techl writing / '20, U Pa, 10, -

Charnes, A / assoc prof math, Carnegie Tech, Pgh, Pa / AM / '17, U Ill, -, -

Cherry, L H / asst to dir, G/M Div, Bu Aer, Navy Dept, Wash, D C / AM / '13, U Cal, 5, engr

Christensen, Veral E / relay engr, Detroit Edison Co, Detroit / ADE / '24, Lawrence Inst Tech, 1, IBM custmr engr

Chu, J C / sr sci, Argonne Natl Lab, Lemont, Ill / ADE / '19, U Pa, 9, engr

Church, Randolph / prof math & mech, US Nav PG Sch, Monterey, Cal / M, analog comp / '04, Yale (PhD'35), 14, prof

Cillié, Charl D / sales engr, General Elec Co, Schenectady, N Y / AS / '06, MIT, 7, sales engr

Cleeton, Claud E / supt, Radio I, Naval Res Lab, Wash, D C / E / '07, U Mich, -, elec sci

Clenshaw, C W / staff, Math Div, Natl Physical Lab, Middlesex, England / AM / '26, London U, 8 -

Cochran, Todd D, Jr / sr dev engr, Eastman Kodak, Rochester, N Y / ACDE / '22, Lafayette Coll, 7, elecl engr

Cockburn, Curtis D / engr, General Elec Co, Syracuse, N Y / ADE / '26, -, 2, elecnc engr

Cohler, Edmund U / res engr, Dig Comp Lab, MIT, Camb / DEM / '28, Nwn U, 3, engr

Coile, Russell C / staff mbr, Operns Evaluatn Grp, Navy Dept, Wash, D C / A, operns res / '17, MIT, -, -

Colebrook, F M / hd, Elecncs Sec, Natl Physical Lab, Teddington, Middlesex, England / AD / '93, London U, 5, -

Coleman, John F / proj engr, IBM, Pkpsie, N Y / DE / '17, Cornell U, 4, engr

Cook, C H / -, - / AMS / '25, U Tex, -, mathn

Coolidge, Charles A, Jr / res assoc, Harv Compn Lab, Camb / ACD / '23, Harvard, 7, des

Coombs, John M / dev engr, IBM Lab, Pkpsie, N Y / DE / '15, U Maine, 13, elecengr

Concordia, Charles / engr, General Elec Co, Schenectady, N Y / AM, engrg studies, contr sys / '08, Gen Elec Adv Engrg Prog, 15, engr

Conner, William J, Jr / engr, Brown Instr Div, Mnpls-Honeywell Regulator Co, Phila / A, process contr & mngm contr / '23, Villanova Coll, Drexel Inst Tech, 5, engr

Corbin, S W / mgr, Indstl Sales, Apparatus, General Elec Co, Schenectady, N Y / AS / '09, Union Coll, -, -

Corderman, Charles L / sec leader, Digital Comp Lab, MIT, Camb / E, storage / '24, MIT, 4, engr

Costello, George R / proj engr, Chance-Vought Aircraft, Dallas, Tex / EM / -, U Mich, -, mathn

Cottrell, Don E / res asst, WRRC, Ypsilanti, Mich / D / '28, U Denver, 1, elecl engr

Craig, Cecil C / prof math & dir, Statl Res Lab, U Mich, Ann Arbor, Mich / AM, math stat / '98, Indiana U, U Mich (PhD), -, tchg & res

Crane, Hewitt D / elecl engr, Inst for Adv Study, Princeton, N J / DE / '27, Columbia U, 4, engr

Crawford, David J / engr, IBM Engrg Lab, Pkpsie, N Y / ACDE / '22, MIT, 9, elecncs engr

Crossman, Loring P / proj engr, Rem Rand, South Norwalk, Conn / ADE / '92, Haverford Coll, 30, engr

Cryan, James J/partner, Machine Statistics Co, N Y / A, IBM tab serv / '24,Fordham U, 5, executive

Cunningham, Clarence S / foreman elecn, Commonwealth Elec Co, Riverdale, N Dak/ ACDEM / '16, Regent St Tech Coll, London, -, elecn

Curtis, A R / staff, Math Div, Natl Physical Lab, Middlesex, England / AM / '23, London U, 4, -

D: Danzinger, L J / asst vp, Bankers Life, Des Moines, Iowa / A, ins / '07, -, -, executive

David, Richard B / engr, Arga Instrument Co, Southampton, Pa / ACDE / '23, Drexel Inst Tech, 2, mechl engr

DeCoatsworth, C G / elecnc engr, Frankford Arsenal, Phila / DE / -, U Pa, 2, engr

Demuth, Howard B / staff mbr, U Cal Lab, Los Alamos, N Mex / ADEM / '28, U Colo, Stanford U, 3, elecnc engr

Deuth, A F / asst chf engr, Hogan Labs, N Y / ACDES / '12, Milwaukee Sch Engrg, 4, engr

Dickerman, J / sr engr, Control Instrument Co, Bklyn, N Y / DE, fire contr/ '15, U Ky, 2, engr

Diemer, Ferdinand P / res engr, Arma Corp, Bklyn, N Y / AEM, automatic proc / '20, NYU, Bklyn Polytech, 7, elecnc engr

Dietzold, Robert L / res mathn, Bell Tel Labs, Murray Hill, N J / EM / '04, MIT, 5, communications

Dillon, John D / chf, Spec Proj, AF Missile Test Cr, Patrick AF Base, Fla/ ADE/ '11, Marshall, Stetson, U Fla, 5, elecncs sci

Disson, Stanley B / Lt, USAF, Dept of Defense, Wash, D C / ACDE / '28, U Pa, 3, elecnc engr

Dixon, Wilfrid J / prof,Statl Lab, Univ of Oregon, - / AM / '15, Princeton U , 12, tchg & res

Doak, Charles A / sales engr, Weston Elecl Inst Corp, N J / EMS / '26, Fairleigh Dickinson Coll, 0, sales engr

Dobbins, Willis E / engr, Comp Res Corp, Manhattan Beach, Cal / DE / '19, Calif Inst Tech, 6, engr

Dodd, Stephen H, Jr / grp leader, Dig Comp Lab, MIT, Camb / E / '20, MIT, 6,engr

Doroshow, Seymour R / sr elecncs engr, Glenn L Martin Co, Baltimore, Md / EM / '25, U Pa, 10, engrg suprvr

Dresch, F W / statn, Stanford Res Inst, Stanford, Cal / AM / -, U Cal, 10, statn

Drillick, Jack H / -, N Y / ACD / '25, -, 3, mach desnr

Duke, Thomas C / dir, Comp Grp, Meteor Engrg, MIT, Camb / AMS / '24, MIT, 2, applied mathn

GLOSSARY — Section 2: C D, E

(First edition, section 2, cumulative, May, 1953)

The following is a glossary of terms used in the field of computers and automation. It is the purpose of this glossary to report the meanings of terms as used, and not to legislate about them. "Section 1: A, B" of this glossary was published in COMPUTERS AND AUTOMATION, March, 1953, vol. 2 no. 2. Additions, comments, corrections, and criticisms will be appreciated. See the discussion in "Forum" in this issue.

C: card — in computer discussion, usually a card of constant size and shape, adapted for being punched in a pattern which has meaning. The punched holes are sensed electrically by wire brushes or mechanically by metal fingers. Also called "punch card".

card column — usually, one of eighty columns in a punch card into which information is entered by punches.

card feed — a punch card machine or mechanism which feeds cards.

card field — a set of card columns fixed as to number and position, into which the same item of information is regularly entered; for example, purchase order numbers of five decimal digits might be punched regularly into the card field consisting of card columns 11 to 15.

card punch — a machine or mechanism which punches cards.

card reader — a mechanism that causes the information in cards to be read, usually by passing them under brushes or across metal fingers.

card stacker — a punch card machine mechanism that stacks cards after they have passed through the machine. Also called "card hopper".

cathode ray tube — in computer discussion, a large electronic vacuum tube containing a screen on which information may be stored by means of the presence or absence of spots bearing electrostatic charges. The capacity usually is from 256 to 1024 spots.

channel — (1) in magnetic tape, a line parallel to the edge of the tape along which information may be stored by means of the presence or absence of magnetized or polarized spots; (2) in a machine, a path along which information may flow; for example, in the machine known as a punch card reproducer, information (in the form of punch cards) may flow in either one of two card channels which do not physically connect.

character — in computing machinery, a decimal digit 0 to 9, or a letter A to Z, either capital or lower case, or a punctuation symbol, or any other single symbol which a machine may take in, store, or put out.

check digits — one or more digits carried along with a machine word (i.e., a unit item of information handled by the machine), which report information about the other digits in the word in such fashion that if a single error occurs (excluding two compensating errors), the check will fail and give rise to an error alarm signal. For example, the check digit may be 0 if the sum of other digits in the word is odd, and the check digit may be 1 if the sum of other digits in the word is even.

coded decimal — a form of notation by which each decimal digit separately is converted into a pattern of binary ones and zeros. For example, in the "8-4-2-1" coded decimal notation, the number twelve is represented as 0001, 0010 (for 1, 2) whereas in pure binary notation it is represented as 1100. Other coded decimal notations are known as: "5-4-2-1", "excess three", "2-4-2-1", etc.

coder — a person who translates a sequence of instructions for an automatic computer to solve a problem into the precise codes acceptable to the machine.

code — (1) express problems in language acceptable to a computer; (2) the precise system of representing information and instructions for a computer. For example, the code 00 may designate the operation addition, the code 01 the operation subtraction, the code 10 the operation multiplication, etc.

coding line — a single command or instruction written usually on one line, in a code for a computer to solve a problem.

collating — combining two sequences of items of information in any way such that the same sequence is observed in the combined sequence. For example, sequence 12, 29, 42 and sequence 23, 24, 48 may be collated into 12, 23, 24, 29, 42, 48.

collator — a punch card machine which has two card feeds, four card hoppers, and three stations at which a card may be compared or sequenced with regard to other cards, so as to determine the hopper into which it is to fall.

comparer — a mechanism by means of which two items of information may be compared in certain respects, and a signal given depending on whether they are equal or unequal.

comparing — testing or examining to see if two items of information are equal or unequal.

compiling routine — a routine by means of which a computer can itself construct the program to solve a problem by assembling, fitting together, and copying other programs stored in its library of routines.

complement — in decimal notation, the number obtained by replacing each decimal digit by nine minus that digit; in binary notation, the number obtained by replacing each digit by one minus that digit. For example, in binary, the complement of 1101011 is 0010100.

complete operation — in a computer, a calculating operation which includes (1) obtaining all the numbers entering into the operation out of the memory, (2) making the calculation, and (3) putting the result back into the memory.

computer — a machine which is able to calculate or compute, i.e., perform sequences of reasonable operations with information.

control unit — in a computer, the unit which controls the sequence of operations in the computer.

converter — a machine which changes information in one kind of language acceptable to a machine into corresponding information in another kind of language acceptable to a machine. For example, a machine which takes in information expressed in punch cards and produces the same information expressed in magnetic tape, is a "converter". Often the machine possesses limited computing facilities, spoken of as "editing facilities".

counter — a mechanism which either totals digital numbers, or allows digital numbers to be increased by additions of one in any column or place of the number.

cycle — in a computer, the smallest period of time or complete process of action that repeats itself in order. In some computers "minor cycles" and "major cycles" are distinguished.

D: debug — remove malfunctioning conditions from a computer (colloquial).

decade — a group of ten; for example, a "decade counter" will count to ten in one column or place of a decimal number.

decimal digit — one of the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 when used in numbering in the scale of ten. Two of these digits, 0 and 1, are of course also binary digits when used in numeration in the scale of two.

decimal point — in a decimal number, the point that marks the place between integral and fractional powers of ten.

decimal-to-binary conversion — mathematical process of converting a number written in the scale of ten into the same number written in the scale of two.

delay line — in a computer, a device which stores information in a train of pulses or waves, and as a pattern of the presence or absence of such waves. An example of a "delay line" in everyday life is an echo; the air and a reflecting wall momentarily store a train of sound waives. In a computer delay line, the medium may be mercury, the container a pipe, and the pulses issuing at the final end may be detected electrically, amplified, reshaped, and reinserted at the beginning end.

differential analyzer — an analog computer designed particularly for solving or "analyzing" many types of differential equations.

digit — one of the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and sometimes also letters, used to designate each of the n-1 numbers smaller than the base n of a scale of numbering.

digital — using numbers expressed in digits and in a scale of notation, in order to represent all the variables that occur in a problem.

digital computer — a computer which calculates using numbers expressed in digits and yeses and noes expressed usually in 1's and 0's, to represent all the variables that occur in a problem.

digitize — change an analog measurement of a physical variable into a number expressed in digits in a scale of notation.

double precision — in computing, computations that uses twice the ordinary length of machine words. For example, in the case of a desk calculator regularly handling ten place decimal numbers, computation with 20 place numbers by keeping track of the 10 place fragments, is computation with "double precision".

down-time — in the operation of a computer, time when it is malfunctioning, or not operating correctly, due to machine failure (colloquial).

E: education of a computer — preparing and assembling programs for a computer so that the computer can itself put together many programs for many purposes. This greatly reduces the time required from human programmers to program the computer.

electric typewriter — a typewriter having an electric motor and the property that almost all the operations of the machine after the keys are touched by human fingers are performed by electric power instead of the power of human fingers and hands.

electronic (as contrasted with "electric") — dealing generally with flows of small numbers of electrons in a vacuum, as contrasted with flows of large numbers of electrons along wire conductors.

electronic calculating punch — a punch card machine produced by International Business Machines (called Type 604), which, in each 4/10 of a second reads a punch card passing through the machine, performs about 50 program steps and punches a result on the punch card.

electrostatic storage — storage of information in the form of the presence or absence of spots bearing electrostatic charges. See "cathode ray tube".

equivalent binary digits — number of binary digits equivalent to a given number of decimal digits or other characters. When a decimal number is converted into a binary number, the number of binary digits necessary is in general equal to about 3 1/3 times the number of decimal digits. In coded decimal notation, the number of binary digits necessary is ordinarily 4 times the number of decimal digits.

erasing -- removing information from storage and leaving the space available for recording new information.

excess-three code -- a coded decimal notation for decimal digits which represents each decimal digit as the corresponding binary number plus three. For example, the decimal digits 0, 1, 8, 9 are represented as 0011, 0100, 1011, 1100, respectively. As may be seen, in this notation, the complement of the decimal digit is the complement of the four binary digits.

external memory -- information stored (in language acceptable to the machine) outside of the machine, as for example, recorded magnetic tape in a closet, or punch cards in filing cabinets.

extracting -- in some computers, the operation of obtaining certain digits from a machine word as may be specified. For example, if the ten digit number 0000011100 is stored in a machine register, the computer can be instructed to "extract" the eight digit (in this case a one) and correspondingly perform a certain action.

1. The Word "Automation". From Nora B. Moser. Alexandria,Va.:

How do you pronounce "automation"? Accent on the second or third syllable? I hope you can clear up this vital matter in the very next issue of COMPUTERS AND AUTOMATION.

Accent is on the third syllable. The word rhymes with "nation".

* * *

2. Reprints. From Harry Winston Mergler, Lewis Flight Propulsion Laboratory, Cleveland, Ohio:

Please send information describing the form in which papers are to be submitted for publication in COMPUTERS AND AUTOMATION. Will there be a per page charge and prints made available as in the "Review of Scientific Instruments"? Congratulations on your undertaking; such a publication represents a long felt need in a field not heretofore adequately covered by any journal.

The form in which papers (or rather articles) are to be submitted to COMPUTERS AND AUTOMATION is explained in the note on "Manuscripts" in each issue. We can easily arrange for any number of reprints at moderate cost. — Our thanks to Mr. Mergler for his good wishes.

* * *

3. Glossary. From C.L. Perry, U.S. Naval Postgraduate School, Monterey, Calif.:

I noted, with interest, your inclusion of part of a glossary of computer terminology in the March 1953 issue of "Computers and Automation". The value of such a glossary depends on the accuracy and completeness of the definitions included in the glossary. One way to assure accuracy would be to have the definitions reviewed by several competent reviewers. This, of course, should be accomplished before publication.

The Association for Computing Machinery has a Committee on Terminology (I believe that Dr. John W. Carr III of the Willow Run Research Center, University of Michigan and Dr. Grace M. Hopper of Remington Rand are members of this committee). Dr. H.D Huskey, Wayne Univ., Detroit, Mich. and others have already compiled glossaries. The Institute of Radio Engineers' Professional Group on Electronic Computers has a Committee on Nomenclature, with chairman Mr. Nathaniel Rochester, IBM Engineering Laboratory, Poughkeepsie, N.Y. In order to avoid conflicting definitions as much as possible it would be desirable for you to compare definitions with these people. It is particularly important for a field for which the vocabulary is growing and changing that glossaries be carefully compiled.

Documentation, i.e., a listing of sources and reviewers, would also be desirable for a glossary.

Some of your definitions appear (to me) to be misleading. For example consider the definition for the term analog. In some mechanical counters rotation is used to represent a digital quantity. In a digital computer a given rotation

represents one value from a finite set of values. In an analog computer a given rotation represents one value from an infinite set of values.

Mr. Perry's viewpoint is important and widely held, and does have some advantages, which his letter makes clear. We believe however that it is in some essential respects both impractical and undesirably authoritarian.

The viewpoint expressed by Mr. Perry is impractical because it is extremely difficult for the several existing committees on terms in the computer field to get together and issue quickly a coordinated report on terms. For example, it is reported that just one of the committees recently spent a whole day of meeting together, in discussing and settling on the definitions to be assigned to only five terms. At such a rate it seems that ten years will pass before their report is finished.

In any new field, terms with new meanings spring up all the time, with different expressions used in different laboratories. It will take some years of wear and tear, and competition, for good terms to be sifted out. Language grows and flowers by its own rules, and these do not often include the deliberations and decisions of committees. Language displays more often a struggle for survival of the fittest words. For example the term "large-scale sequence-controlled calculator" was current in 1947; but by 1953 it had been very largely superseded by the term "automatic computer".

Also, the viewpoint expressed by Mr. Perry is authoritarian, in that the opinion of "authorities" is "sanctified". It leads committees to think that they should establish by fiat the meaning of new terms, instead of simply reporting different usages in different places. For example, in a recently published glossary the term "novendenary" was set down (as a correlate of "binary" and "decimal") to refer to computation in the scale of nineteen. But no computer man in any laboratory to our knowledge has ever considered computation in the scale of nineteen, which would have many conspicuous disadvantages and no advantages as yet conceivable.

The principle that it seems to us should guide the presentation of glossaries in this early stage of the computer field is simple reporting of the meanings attached to terms. If several meanings are attached to one term, or if there are conflicting definitions, then let us report all of them. After all, the people who will refer to the glossary are mainly newcomers to the field; and as soon as they obtain a substantial part of the idea or ideas corresponding to a term, they will proceed to guess the rest of its meaning (and modify its meaning) from the way it is used in the context.

We shall be glad to publish specific comments and criticisms on definitions of terms, and alternative, or better, definitions as soon as we find them. For example, see Mr. Perry's comments on "analog".

MEDICAL DIAGNOSIS

by Dr. Marshall Stone

Professor of Mathematics, University of Chicago

(from a letter to the Editor)

Have computer men ever examined medical diagnosis as a matter of logic? During recent years the "New Yorker" has published two or three articles, based on the difficulties of diagnosis, which made me reflect a little on ways of overcoming such difficulties. It then occurred to me that computer men might have given thought to the subject, or might be inclined to examine it sometime in the future.

While I am entirely ignorant of how diagnosis is taught in the medical schools and hospitals, I cannot be far wrong in supposing that diagnosis is essentially a logical problem rather like that of qualitative chemical analysis. I would be rather surprised if diagnosis were taught in any more rational fashion than qualitative analysis was when I took a course in the latter in 1919-20. Then we were given a systematic procedure which we had to learn and practice until we arrived at a stage of moderate expertness. We were never taught to see in this procedure a logical apparatus designed in the light of a detailed comparison of the chemical properties of the various elements.

Medical diagnosis, however it may be taught, presents some rather serious problems, if the situations one hears of from time to time are any guide. Many diseases are probably not discussed at any length, just as in my course in qualitative analysis no attention was paid to the rare earths and various other elements. Many doctors never see a case of a once common disease, like smallpox, for instance, and some prove to be incapable of identifying it when it crops up unexpectedly in practice. In some circumstances the diagnostic procedures seem to be uncertain or over-delicate, with the result that wrong diagnoses in rather surprising variety are not uncommon. All this leads to unnecessary suffering and even death, not perhaps on a statistically alarming scale but nevertheless to a degree which cannot be regarded as satisfying from a scientific point of view.

It would be too much to expect every doctor to have an extremely thorough training in diagnosis or to gain in practice the exceptional skill which some doctors certainly develop as diagnosticians. However, it might be possible to do two things: first, revise the basic doctrine and reform its teaching in the light of modern logic; and then devise highly refined procedures (probably mechanized) which could be used by especially staffed diagnostic centers, to which difficult or uncertain diagnoses could be referred on a consulting basis.

If the computer fraternity should get interested in this sort of thing, could find support for their research, and should study the whole problem, I believe that they might be able to perform a very useful service.

AUTOMATIC COMPUTERS — LIST

(Edition 2, supplement, information as of May 3, 1953)

The purpose of this list is to report automatic computers in existence (all that are known to us). Each entry gives: name of computer (and interpretation of letters) / name of maker, place / purpose of computer, nature of computer, approximate size or capacity of computer, and quantity of computer in existence. Some words like "Model" and "Type" have been omitted from names of computers; usually the initial letters of the company name have been substituted.

We plan to keep this list up to date from time to time, and we shall be very grateful for any information which any reader is able to send us.

Although we have tried to make this list complete and accurate, we assume no liability for any statements expressed or implied.

This edition contains only revisions or additions as compared with Edition 1, cumulative, published in the March, 1953, issue of COMPUTERS AND AUTOMATION vol. 2, no. 2.

Abbreviations: The key to the special abbreviations follows:

Purpose (p)	Size (s)
Gp General purpose	Ss Small size or low capacity
Sp Special purpose	Ms Medium size or medium capacity
	Ls Large size or large capacity
Nature of computer (c)	
Dc Digital computer	**Quantity (q)**
Ac Analog computer	0q Zero (i.e., unfinished or dismantled)
Ec Electronic computer	1q One
Rc Relay computer	2q Two
Mc Mechanical computer	Sq Small quantity, about 2 to 6
	Mq Medium quantity, about 7 to 30
	Lq Large quantity, over 30
	?q Unknown quantity

Some other abbreviations have been used which can be easily guessed, like those in a telephone book.

ANSER (Analog Simulator and Computer) 300-A / Davies Laboratories, Inc, Riverdale, Md / Sp EAc ?s ?q
Computer / Electronics Div, AERE, Harwell, England / EDc
Computer / Imperial College, Univ of London, London / Gp RDc ?s 1q
Computer / Naval Special Devices Center, Pt Washington, N Y / Dc
GEDA (Goodyear Electronic Differential Analyzer) L2, L3, N3 (linear and nonlinear models) / Goodyear Aircraft Corp, Akron, Ohio / Gp EAc ?s ?q
IDA Electronic Slide Rule ((listed previously only as "Computer")) / Computer Corp of America, N Y / Gp EAc ?s ?q
Kalin-Burkhart Logical Truth Calculator / T A Kalin & W Burkhart, Cambridge, Mass / Sp Rc Ss 0q
Manchester Computer / Univ of Manchester, England / Gp EDc Ls 1q
Remington-Rand 409 Computer / Remington Rand, N Y / Gp EDc Ss ?q
Statac (Statistical Automatic Computer) / National Bureau of Standards, Washington, D C / Sp Dc ?s 1q